Conception and Validation Software Tools for the Level 0 Muon Trigger of LHCb

J. Cogan, J. P. Cachemiche, P. Y. Duval, F. Hachon, E. Aslanides, A. Tsaregorodtsev, R. Le Gac, O. Leroy, P. L. Liotard, F. Marin

Conception and validation software tools for the level 0 muon trigger of LHCb

E. Aslanides, J.-P. Cachemiche, J. Cogan, P.-Y. Duval, R. Le Gac, F. Hachon, O. Leroy, P.-L Liotard, F. Marin and A. Tsaregorodtsev,

Centre de Physique des Particules de Marseille,

Aix-Marseille Université, Marseille, France, on behalf of the LHCb collaboration

Abstract—**The Level-0 muon trigger processor of the LHCb experiment looks for straight particules crossing muon detector and measures their transverse momentum. It processes 40×10⁶ proton-proton collisions per second. The tracking uses a road algorithm relying on the projectivity of the muon detector. The architecture of the Level-0 muon trigger is complex with a dense network of data interconnections. The design and validation of such an intricate system has only been possible with intense use of software tools for the detector simulation, the modelling of the hardware components behaviour and the validation. A database describing the dataflow is the corner stone between the software and hardware components.**

***Index Terms*—First level trigger, Muon Detector, LHCb**

I. Overview

The LHCb experiment [1] is installed at the Large Hadron Collider at CERN, to study CP violation and rare decays in the beauty sector.

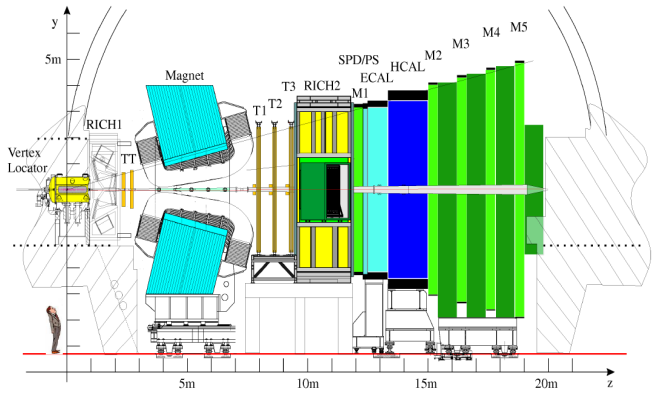

Fig. 1: The layout of the LHCb spectrometer.

Interesting b-hadron decays in proton–proton collisions at a centre of mass energy of 14 TeV have to be isolated in a large background. The LHCb detector, shown in Fig. 1, is a single-arm spectrometer covering the forward region of the proton–proton interactions. It is composed of a vertex locator, two Ring Imaging Cherenkov counters, a warm dipole magnet producing a vertical field, two groups of tracking stations before and after the magnet, a preshower, an electromagnetic and hadronic calorimeters and a muons detector.

The LHCb trigger is divided into two systems: a Level-0 trigger and a high level trigger (HLT). The purpose of the Level-0 trigger is to reduce the LHC beam crossing rate from 40 to 1 MHz where the entire detector can be read out. The Level-0 trigger is based on custom electronics collecting dedicated information from the pile-up detector, the calorimeters and the muon detectors. The latency to process a proton-proton collision is limited to 4 µs. This time includes the particules time-of-flight, cables lengthes and all delays in the front-end electronics, leaving ~2 µs for the processing of the data in the Level-0 trigger. The purpose of the HLT is to reduce the rate down to 2 kHz by using data from all subdetectors. It is based on a farm of about one thousand computers interconnected through a gigabit Ethernet network.

The Level-0 muon trigger is described in [2]. In this section, we give an overview of the muon system and of its achitecture. In section II, we present in details the software tools developed to design, debug and validate this very complex system. The software model of the Level-0 trigger processor is explained in section III, while the debugging and validation methodologies are covered in section IV.

A. Muon detector

The muon system [1] has been designed to identify muons with a high transverse momentum: a typical signature of a b-hadron decay. Muons are presented in the final states of many *CP*-sensitive b-meson decays and provide a tag of the initial flavour of b-mesons. The muon system is divided into two intimately related subsystems: the muon detector and the Level-0 muon trigger.

The muon detector, shown in Fig. 2 consists of five muon stations interleaved with muon filters. The first filter,

between station M1 and M2, is naturally provided by the electromagnetic and hadronic calorimeters. The muon filters following M2 to M5 stations are made of iron absorber blocks. The muon stations provide binary space point measurements of the tracks. They are segmented in pads that are narrower in the magnet bending plane to give an accurate measurement of the transverse momentum, p_T. The segmentation is projective to ease the tracking in the Level-0 muon trigger. The pad size depends on the station and on its location in the station. Along the *x*-axis, the segmentation is twice finer for M2-M3 and twice coarser for M4-M5 with respect to M1.

Stations M2-M3 are devoted to the muon track finding while stations M4-M5 are used to confirm the muon identification. The first station M1 is placed in front of the calorimeter and plays an important role for the p_T measurement of the muon track.

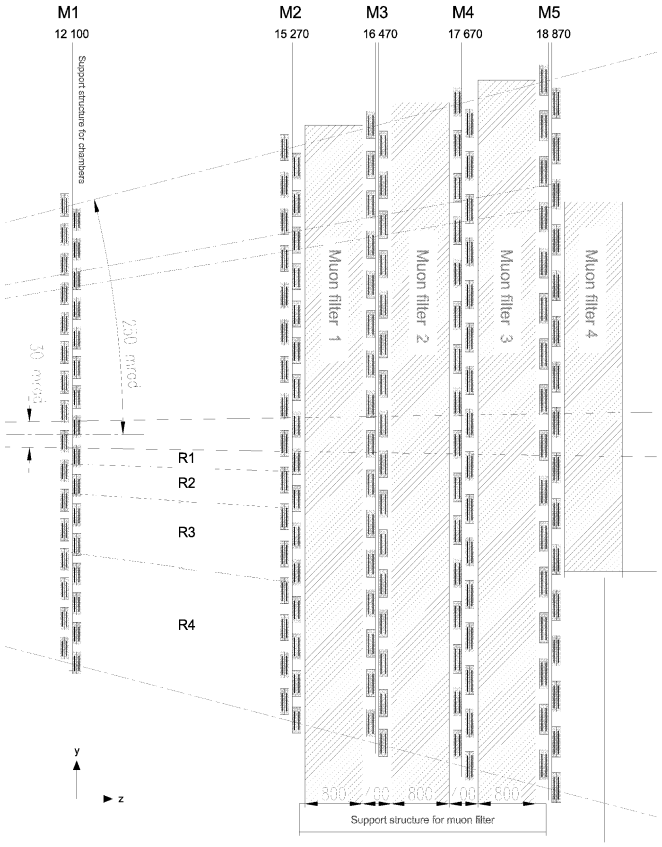

Fig 2: Side view of the muon system in the y, z plane showing the iron filters, stations and chamber organization as well as the projectivity of the system.

Each station is subdivided into four regions with different logical pad dimensions as shown in Fig. 3. Region and pad sizes scale by a factor two from one region to the next. The logical layout in the five muon station is projective in *y* to the interaction point. It is also projective in *x* when the bending in the horizontal introduced by the magnetic field is ignored.

Pads are obtained by the crossing of horizontal and vertical strips where applicable. Strips are employed in stations M2-M5 while station M1 and the inner most region, R1, of station M4-M5 are equipped with pads. The Level-0 muon trigger receives 25,920 logical channels every 25 ns forming 55,296 logical pads by crossing strips.

Each region is subdivided into *sectors* as shown in Fig. 3. They are defined by the size of the horizontal and vertical strips and match the dimension of underlying chambers.

B. The level-0 muon trigger

The Level-0 muon trigger looks for muon tracks with a large p_T. The track finding is performed using the logical pad information. It searches for hits defining a straight line through the five muon stations and pointing towards the interaction point, as shown in Fig. 5. The position of the track in the first two stations allows the determination of its p_T.

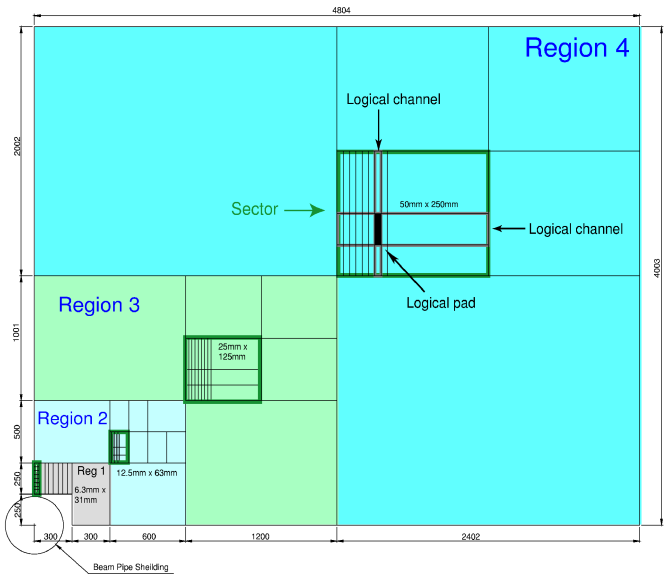

Fig 3: Front view of one quadrant of muon station M2, showing the dimension of the regions. Inside each region is shown a sector, defined by the size of the horizontal and vertical strips

To simplify the processing and to hide the complex layout of stations, the muon detector is subdivided into 48×4 = 192 *towers* pointing to the interaction point. The tower organization is shown for a quadrant of the muon detector in Fig. 4.

All towers contains logical pads with 48 pads from M1, 96 pads from M2, 96 pads from M3, 24 pads from M4 and 24 pads from M5. Therefore, the same algorithm can be executed in all towers. Each tower is connected to a processing unit (PU), the key component of the trigger processor.

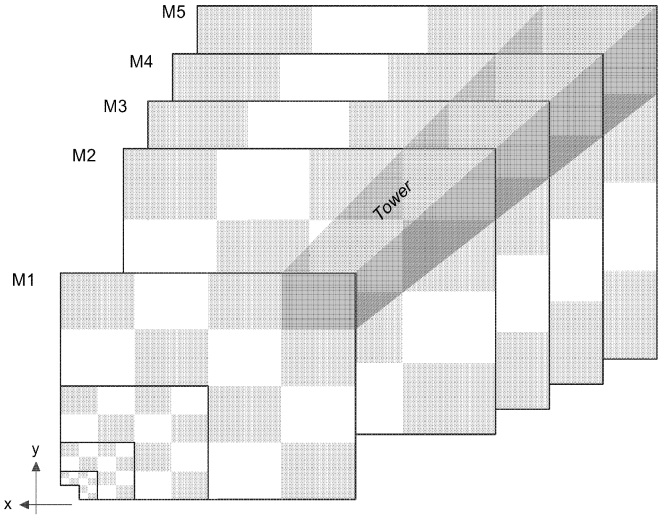

Fig 4: A quadrant of the muon detector with its 48 towers pointing towards the interaction point

C. The track finding algorithm

The track finding is based on a road algorithm illustrated in Fig. 5 and 6. It assumes muon track coming from the interaction point with a single kick from the magnet.

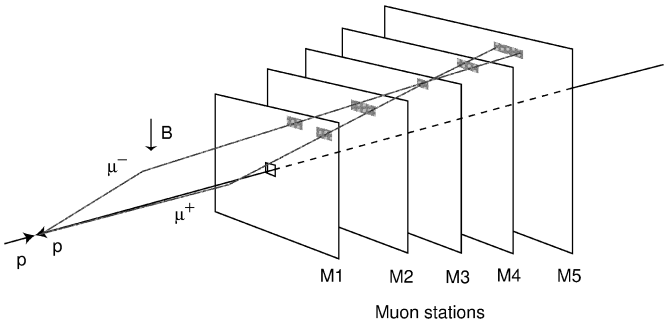

Fig 5: Track finding by the Level-0 muon trigger. In the example shown, μ+ and μ- cross the same pad in M3. Grey areas illustrate the field of interests used by the algorithm for station M1, M2, M4 and M5.

For each logical pad hit in M3, the straight line passing through the hit and the interaction point is extrapolated to M2, M4 and M5. Hits are looked for in these stations within search windows, called field of interest (FOI), approximately centred on the straight line extrapolation. The size of the FOI depends on the considered station, the distance from the beam axis, the level of background and the minimum-bias required retention. When at least one hit is found inside the FOI for each of the stations M2, M4 and M5, a muon track is flagged and the pad hit in M2 closest to the extrapolation from M3 is selected for a subsequent use.

The track position in station M1 is determined by making a straight-line extrapolation from M3 and M2, and identifying in the M1 FOI the pad hit closest to the extrapolation point.

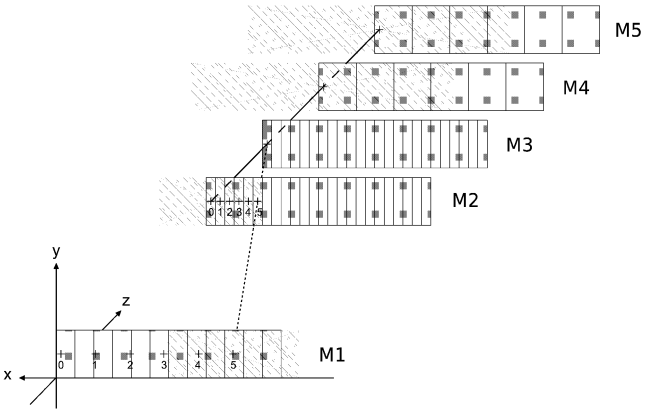

Fig 6: Fields of interest associated with an M3 pad located on the left border of a tower. The straight line shows the extrapolated position in stations M2, M4 and M5. The hatched areas show the maximum size of the field of interest centred on the extrapolated position where hits are searched. Has an example the dotted line shows the extrapolation from M3 and M2 to M1 to the pad labelled 5, hit in M2. Numbers show the one-to-one correspondence between a pair of pads hit in M3–M2 and the extrapolated position in M1. The track finding is also performed along the y-axis for station M4 and M5. The y-size of the field of interest is 1 pad. It is not drawn for simplicity

Since the logical layout is projective, there is one-to-one mapping from pads in M3 to pads in M2, M4 and M5. There is also a one-to one mapping from pairs of pads in M2 and M3 to pads in M1. This allows the track finding algorithm to be implemented using only logical operations.

The implementation of the track finding algorithm is complex: it involves a large number of logical channels distributed in a large volume; a mixture of pads and strips; and segmentation of logical channels varying between regions and stations. There is a one-to-one correspondence between towers and trigger sectors except for region R1 of station M2-M3, where a trigger sector is shared by two towers, and in region R2 where a tower maps two sectors, as shown in Fig. 3.

Finally, each PU has to exchange a significant amount of data with its neighbours to avoid inefficiency at the border of a tower. The quantity of exchanged logical channels is determined by the maximum width of fields of interest, and depends on the location of the PU. A PU emits 224 and receives 214 logical channels in the worst case.

E. The data acquisition system

The Level-0 muon trigger, as the others components of the experiment, is interfaced to the LHCb data acquisition system, shown in Fig. 7 [1].

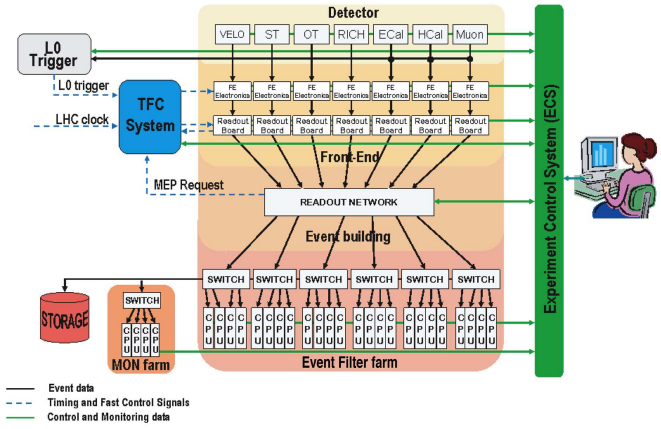

Fig 7: The LHCb data acquisition system and its connections to the Level-0 trigger system, TFC system and the ECS system.

It is composed of a set of standardized readout boards, the Tell1 boards [3], a gigabit Ethernet network, the event filter farm, some storage and the Timing and Fast Control system (TFC). For each accepted collisions, each detector component sends data fragments through the Tell1 boards where they are merged, formatted, compressed and embedded into IP-packets. The latters are pushed into a large switching network providing the connectivity between the Tell1 boards and the individual farm nodes where the event building, high level filtering as well as data reconstruction are performed. The maximum readout rate of the data acquisition system is 1.1 MHz.

II. Architecture

An overview of the Level-0 muon trigger processor architecture is given in Fig. 8. Each quadrant of the muon detector is connected to a separate Level-0 muon trigger processor through 312 optical links transmitting a 32-bit word every 25 ns by serializing data at 1.6 Gbps.

A. Crate and boards

A Level-0 muon trigger processor is a 9U crate containing 12 processing boards, one controller board and a custom backplane. It is connected to the Level-0 decision unit which collects all muon candidates, to the data acquisition system of the experiment and to the TFC.

A processing board houses five processing elements, four PUs and one BCSU (Best Candidate Selection Unit). A PU runs 96 tracking algorithms in parallel, one for each M3 pad of the tower, while the BCSU selects the two muons with the highest momentum within the board.

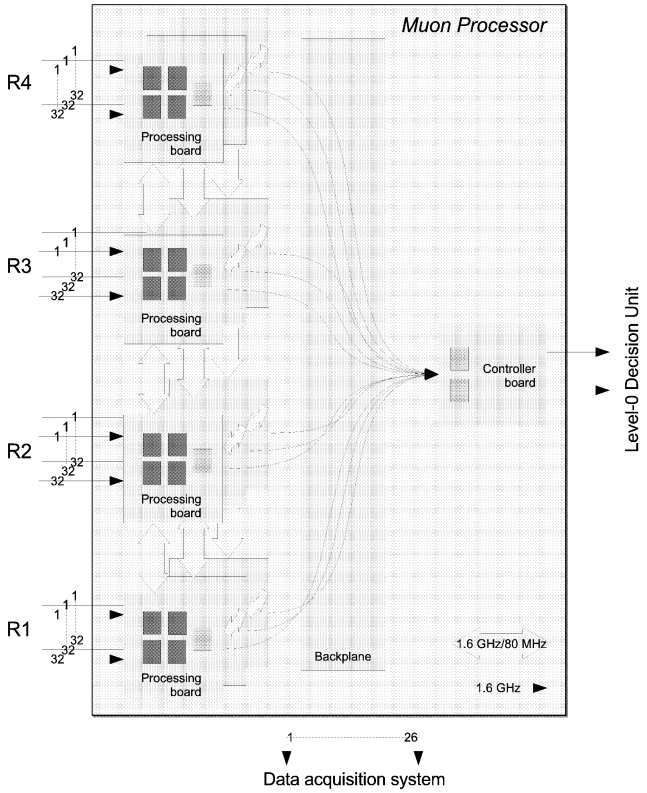

Fig 8: Overview of a Level-0 muon processor

A controller board houses a *control unit* and a *slave unit*. They receive 24 candidates from 12 processing boards, select the two with the highest p_T and send them to the Level-0 decision unit. This board also distributes the 40 MHz clock and the TFC signal via the backplane.

The custom backplane is necessary for the exchange of logical channels between PUs located on different boards and to distribute the master clock and the TFC signal.

All logical channels belonging to a tower are sent to a PU using a maximum of eight optical links: two for station M1, one or two for M2, one or two for M3, one for M4 and one for M5.

The trigger implementation is massively parallel, pipelined and fully synchronous with the LHC clock. It relies on 248 high density Field Programmable Gate Arrays (FPGAs) and on the massive used of multigigabit serial links transceivers embedded inside the FPGAs.

A single generic processing board was designed. The size of each connection between PUs has been maximized to accommodate all configurations. We use 40 MHz and 80 MHz parallel links as well as 1.6 GHz serial links. The

resulting topology of the data exchange is shown in Fig 9. In such design, firmwares loaded in FPGAs differ according to the area covered by the board. We handle 48 configurations, one per tower of a quadrant.

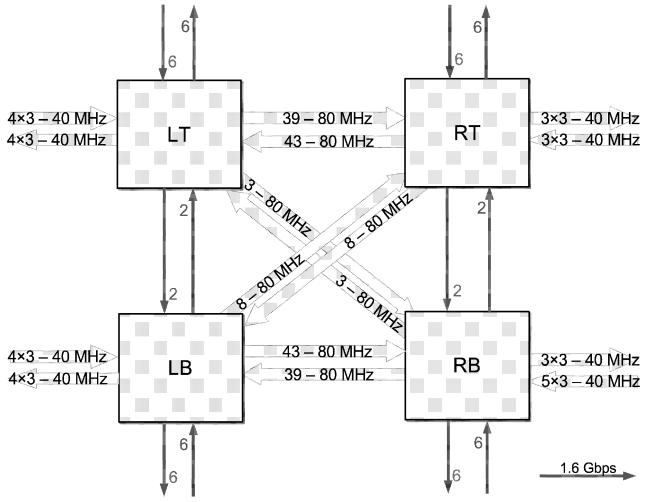

Fig 9: Interconnections between PUs located on a processing board. They are based on 40 MHz parallel links, 80 MHz double data rate links and 1.6 Gbps serial links. Links on the periphery are used to connect PUs located on different boards. The formula in the arrows, describes the number of parallel links connecting PUs located on different boards, the width of the data path for each parallel link and the frequency of the data transfer.

B. Level-0 buffers

The processing and controller boards keep input and output values at different steps of the processing, inside internal memories called *Level-0 buffers*. The content of each Level-0 buffer is sent to the DAQ via six Tell1 boards, for accepted collisions. They provide the key source of information to monitor and debug the Level-0 muon trigger processor.

In the processing boards a copy of the data received from the muon detector are saved with the error status of the optical links. Data exchanged between neighbouring PU are also stored. Finally, the candidates found by each PU and those selected by the BCSU are written .

In the controller boards, information on each pair of candidates sent by each processing boards is stored with the error status of the backplane transmission links, as well as with the final two best candidates.

The width of Level-0 buffers varies between 352 and 720bits resulting in an event size of about 3,5 kBytes for one processor. Zero suppression and data compression algorithms applied later in the Tell1 boards reduce the event size by a factor 10.

III. Software model of a processor

The level-0 muon trigger has a compact architecture with a very complicated data flow. We handle this complexity with the development of a series of dedicated software tools. The core component of the model is a relational database describing the map of exchanges for each logical channel.

In such a system, any error or malfunction can be difficult to identify, understand and interpret. We therefore developed an emulator which reproduces at the bit level the behaviour of each processing element.

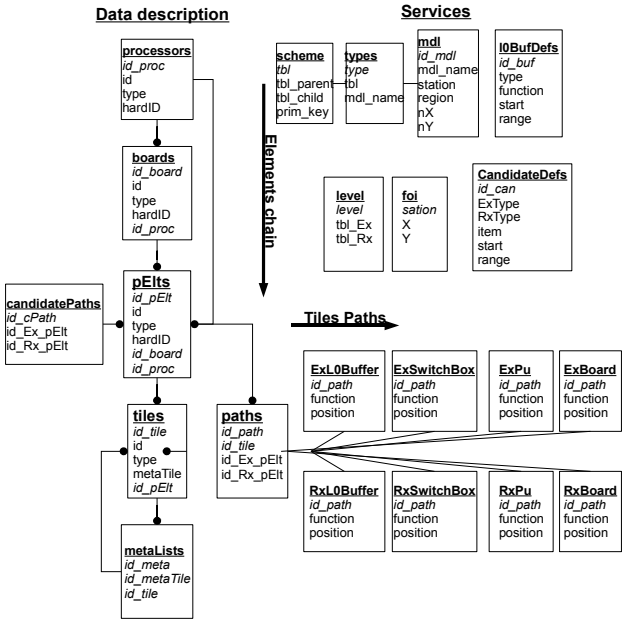

Fig 10: The shema of the database

Finally, dedicated tools are used to debug, validate and monitor the functionality of the hardware. In the following section the basic components of this software are described

A. MuonLayout and MuonTileID

Two software generic objects have been designed to identify and manipulate any element used in the muon system: the *MuonLayout* and the *MuonTileID*. The *MuonLayout* describes a grid by defining its horizontal and vertical dimensions. The muon detector can be fully described by a set of 20 *MuonLayout*: one per station and per region. Each pad or strip can be localized unambiguously by specifying the quadrant, the station, the region, the *MuonLayout* and the *x*, *y* coordinates inside this layout. The *MuonTileID* stores these informations, encoded as a single integer.

More generally, *MuonLayout* and *MuonTileID* are used to represent any element of the Level-0 muon trigger: a

processing board, a processing unit as well as the logical channels.

The *MuonLayout* and the *MuonTileID* implement a set of methods that allow to perform all necessary operations such as inclusions, intersections or finding neighbours. Inclusion operations enable to overlay two layouts and to retrieve all tiles from a given layout in a *MuonTileID* of the other layout. A typical intersection operation is the crossing of logical channels to build a logical pad. The *MuonLayout* and the *MuonTileID* are the elementary bricks on which all the level 0 muon software was build.

B. The database

The database is organized around two main concepts: the *elements chain* and the *tiles paths*, as shown in Fig. 10.

The *elements chain* describes the various components: processors, boards, processing elements or tiles[1]. These components form a hierarchical structure that is reflected in the database by the table relations. Each component is identified by its *MuonTileID*.

The *tiles path* depicts the locations where the tiles can be found within the Level-0 muon trigger processor. A *path* is a set of locations taken by a tile when travelling from an emitting processor element to a receiving processor element. A location is determined by a *function* and a *position* at a given level. The database contains 12914 paths.

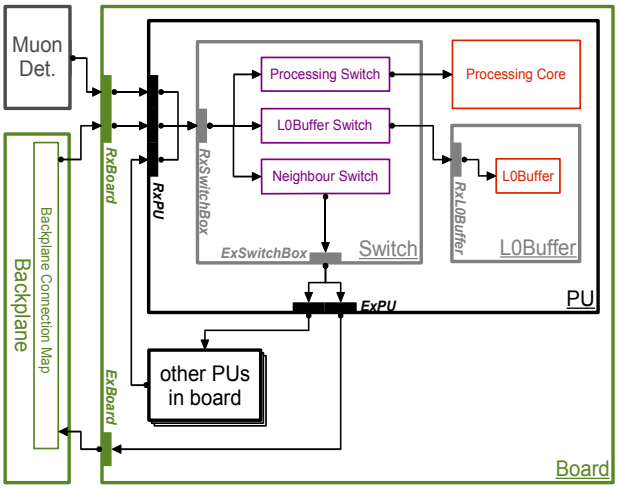

Fig 11: The different possible way for a path flow inside the processor as they are modelled in the database.

Seven levels are necessary to describe all aspects of the Level-0 muon processor. They are shown in Fig. 11. The *ExPU* and *RxPU* levels are related to the PU and describe the tile transfer between two PUs using the logical functions: *Horizontal*, *Vertical*, *Crossing* and *Backplane*. The function characterizes the type of exchanges within a board or between two boards. These two levels were used during the conception of the Level-0 muon trigger to size properly the input and output of each FPGA. They are also used to configure the emulator. The *ExBoard* and the *RxBoard* levels are related to the processing boards. The associated functions identify the connectors of a board. They have been used for the CAD of the backplane. The *ExSwitchBox*, *RxSwitchBox* and *RxL0Buffer* levels are related to the programming of the PUs and are designed to enable the automatic generation of internal switch-boxes. The latter are required to specialize the PU firmware since they all embed identical input/ouput modules and identical core despite the fact that all PU receive and generate a particular data flow. Three kind of switch-boxes are extracted from the database. The first one connects optical link input ports to neighbouring PU output ports. The second one links input ports to the Level-0 buffer and the last one links input ports to the generic computing core.

During the design phases the database was continuously updated to keep an updated image of the processor organization. The *element chain* is naturally fed using the *MuonLayout* of the various components. The *tiles path* is completed using some external tables documenting the optical link organization and some rules enacted to arrange the data exchange. For each processor, 12914 paths are described in the database. This database is designed along a relational implementation model. We use the SQLite [5] library implementing a self-contained, serverless, zero-configuration, transactional SQL database engine which is in the public domain. The interface to this database have been developed as C++ and Python libraries.

C. Emulator of the Level-0 muon trigger

The emulator is a software reproducing on a bit-to-bit basis the *Level-0 buffer* of each processing element. Thus, it allows to check every steps of the processing by comparing the emulated *Level-0 buffers* to the ones produced by the hardware. In that way, we can ensure in particular that the complex data flow of the Level-0 muon system is under control.

The emulator is implemented in C++ and is based on two generic classes: the *unit* and the *register*. A register is a bank of data identified by a name, centrally controlled by a register factory. A *unit* is a simple object manipulating registers. It applies a function on a set of *input registers* and fills a set of *output registers*. It may also contains other units and trigger the execution of their function.

To simulate the Level-0 muon trigger, units are specialized to represent a processor component such as a processing board or a processing element. The emulator is configured with the database that describes the processor topologies. It then forms a hierarchical system of units that communicates through a set of formatted registers reproducing exactly the data transferred inside the processor.

[1] A *tile* refers to a logical channel or logical pad or to a group of logical channels or pads

IV. Debugging and validation tools

The emulator and Level-0 buffers are the key components to debug the system and to validate its functionality, at any time, during a data taking period or later on in the ultimate phase of data analysis. To make an effective use of it, we have developed specific tools.

A. Boards embedded test features

Each board has a CCPC (credit-card PC) embedded, running Linux and interfaced to FPGAs by a custom 16-bits bus. In this way, the operation of any FPGA of the system is controlled and monitored through error detection mechanisms, error counters, spy and snooping mechanisms.

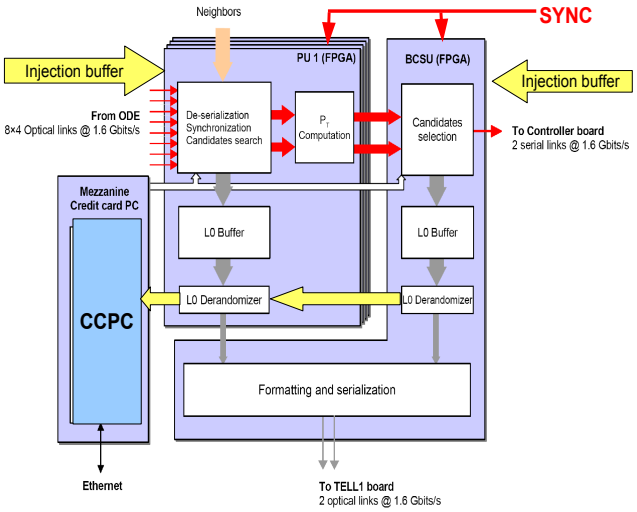

Fig 12: Hardware implementation of the processing board where the interfaces with the CCPC and the DAQ are shown.

Dedicated buffers, shown in Fig. 12, are implemented in PUs to ease the debugging and hardware validation. Injection buffers push data in the processing chain at a rate of 40 MHz replacing the data coming from the optical links. These buffers have a depth of 3563 events and are interfaced to the CCPC. The Level-0 buffers are connected to the data acquisition system of the experiment and are read at a high frequency of 1.1 MHz. Using these buffers to debug the hardware requires a fully functional data acquisition system since the Level-0 buffers can not be accessed by the CCPC.

To allow stand alone tests, a spy mechanism has been implemented. It deactivates the sending of the buffers to the acquisition and permits the CCPC to access the Level-0 buffers of 16 consecutive events. The reading by the CCPC is slow and allowing a maximum rate of about 50Hz.

Above theses buffers, a software running on a distributed architecture has been developed to implement various kind of tests. As shown in Fig 13, it is composed of a master PC taking care of the global sequencing of actions in a distributed and synchronized manner. The CCPC acts as slaves executing local actions on processing and controller boards. The master provides a graphical user interface to run the tests. It also presents a report to the user. Commands and global sequencing of actions are implemented using TCP/IP XML messages while data are shared by the master and the CCPC via a shared disk system using the Network File System protocol. Most of the code is written in Python.

B. Validation methodology

The aim of the validation is to verify that the hardware satisfies its specifications. Usually the specifications are sets of diagrams or documents which explain how to use the system, the expected behaviour and the produced outputs. In our case, the bit-to-bit emulator is the specification model for the Level-0 muon trigger. The hardware processor is validated once it produces the bit-to-bit emulator outputs using the same input data.

We designed different types of validation tests to clearly separate the pure hardware issues from the firmware ones. The first one addresses the hardware layer and verify that the interconnection matrix between processing elements is performing well. The second one is devoted to the higher level functionality, namely the Level-0 muon trigger processing. The last one, validates the stability of the system on a long period of time as well as the data acquisition chain.

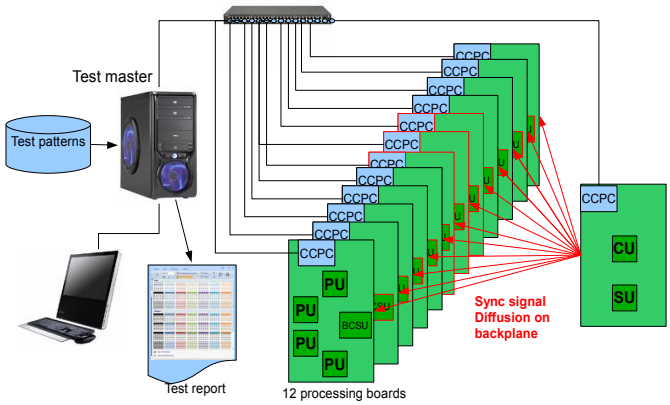

Fig. 13: The test architecture with a master PC and 13 CCPC slave units.

C. Hardware validation

The hardware validation test verifies that all internal links between processing elements are running well with a bit error rate satisfying the requirements.

FPGAs are loaded with a unique dedicated firmware . A 2048 words frame is emitted continuously on each link. The first 12 words of the frame form the header, containing the address of the emitter. Data words of the frame merge several 6-bit counters to fill the width of the link. On the receiving side, each link is synchronized using the header words, extracts the emitter identification and checks that the received data words corresponds to the expected ones. Erroneous words are counted up and published at the end of the test. For this test, the identification data and connection data are provided in tables that have been extracted form the database. All 4 Level-0 muon trigger processors have been validated with at least 12 hours long test without any errors (about 10^{-16} bit error rate).

D. The functional test

The functional test validates the functionality of a Level-0 muon trigger processor covering the VHDL programming of the track finding as well as the internal time alignment mechanism.

Monte-Carlo events are generated and processed by the bit-to-bit emulator in order to produce the input and output patterns. In order to achieve a large coverage of the possible tracks finding, we use a so called “muons gun” which scans randomly the detector acceptance, generating high density muon tracks in all the phase space.

Injection and spy mechanism are used in this test to load input patterns and to collect processing outputs. The master asks each CCPC to load data into their injection buffers, to time align the processing and controller boards and to launch the trigger processing. Finally, the master collects output for each individual board and checks the results against the emulator ones. Injection buffers are loaded with 3563 events. They are processed by groups of 16, limited by the capacity of the spy mechanism. Once all the 3563 events have been treated, the injection buffers are loaded again with new events.

We produced a set of 1 million events stored in the file system. We used them to validate the Level-0 muon trigger processing. They are reused after shutdown periods or each time new firmwares are produced. Although one million events are processed in 25 milliseconds in normal running conditions, we need 6 hours to run this test.

The first version of the CCPC code was written in Python and then rewritten in C which is faster to execute.

E. Stability test

. The stability test uses the complete read out chain. It allows to validate the data processing in the TELL1 and to study the stability of the Level-0 muon trigger during a long period of time. During this test, the system can be operated at its nominal readout frequency of 1.1 MHz.

The injection buffers are loaded once for ever with 3563 events and read in a circular way. This feature is used in real data taking conditions with the LHCb data acquisition systems running. It is thus possible to have days of acquisition with known events that can be sampled by random triggers generated by the TFC. Recorded data are checked offline to verify that input patterns correspond to the generated one and that processing outputs match the emulator one.

V. Conclusion

The Level-0 muon trigger required an innovative architecture to handle the high input rate, the complex data flow and the large volume of data. It relies on a large number of high speed optical links, high density FPGAs and high speed serial links between FPGAs. Such a system could not have been designed and validated without the help of powerful software tools to master its complexity.

We created a versatile software model describing detector sensors layout, the hierarchical organization of the processor components and the mapping between detector sensors and processor components. From this model, we generate the relational database used to configure the software emulator of the Level-0 muon trigger and extract the implementation data required by our electronic CAE tools.

The emulator fully specifies the Level-0 muon trigger. It mimics the hardware behaviour on a bit-to-bit basis. It generates reference patterns to test and debug the hardware. A distributed software was developed to validate the Level-0 muon trigger with test patterns and produce the necessary reports to help for diagnostic. Complementary tools allow to check and to validate step by step, at any time, the three layers of the Level-0 muon trigger: hardware and communication, high level firmware and stability in operational environment.

We dealt with the complexity by creating a series of softwares giving us a continuous chain from the early design and optimization phase up to the processor implementation and operational validation.

The Level-0 muon trigger is operational and was used in order to select many cosmics events as well as the few events delivered during the first LHC start-up period in September 2008.

VI. References

[1] The LHCb Collaboration, *The LHCb detector at the LHC,* JINST **3** (2008) S08005

[2] E. Aslanides et al, *The Level 0 muon trigger for the LHCb experiment*, Nucl. Instrum. Meth. **A579** (2007) 989–1004

[3] G. Haefeli et al, *The LHCb DAQ interface board TELL1*, Nucl. Instrum. Meth. Res. **A560** (2006) 494

[4] The SQLite web site: http://www.sqlite.org